

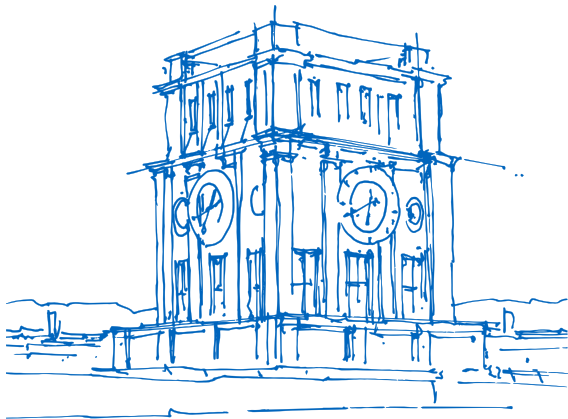
Übung 07: Sequenzielle Logik

Einführung in die Rechnerarchitektur

Michael Morandell

School of Computation, Information and Technology
Technische Universität München

2. – 8. Dezember 2024



Montags:

<https://zulip.in.tum.de/#narrow/stream/2668-ERA-Tutorium—Mo-1000-4>



Donnerstags:

<https://zulip.in.tum.de/#narrow/stream/2657-ERA-Tutorium—Do-1200-2>



Website: <https://home.in.tum.de/momi/era/>

Keine Garantie für die Richtigkeit der Tutorfolien.
Bei Unklarheiten/Unstimmigkeiten haben VL/ZÜ-Folien recht!

- Quiz
- Wiederholung
- Tutorblatt
 - ☐ Wellenformen
 - ☐ Program Counter
 - ☐ Linear-Rückgekoppeltes-Schieberegister

Zitat der Woche

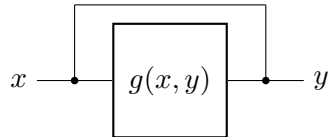
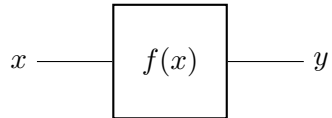
“Also ich finds auch nach dreißig, fünf- nach sechzig Jahren in denen ich diese blöde Vorlesung hier halte, ich finde es immer noch so geil”

- Prof. Dr. Robert Wille (geboren 11.11.1982)

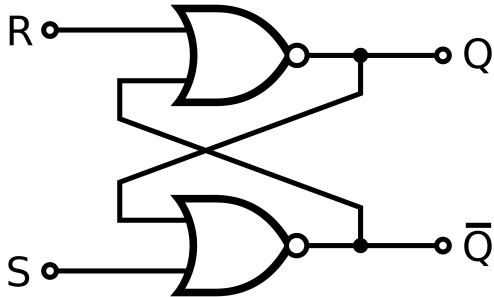
Quelle: [Lecture: November 19. 2024 \(tum.live\)](#)

Sequentielle Schaltungen

- **kombinatorische** Schaltungen: zustandsfrei, Ausgänge nur abhängig von Eingängen.
→ z.B.: HA letzte Woche, Addierer, XOR, ...
- **sequentielle** Schaltungen: zustandsbehaftet, Ausgänge wirken über Rückkopplung auf Schaltung ein! (Zyklus im Graphen)
→ z.B.: Zähler, Speicher, Statusautomaten,
- sequentielle Schaltungen ermöglichen es erst, komplexe Dinge wie Prozessoren zu bauen!



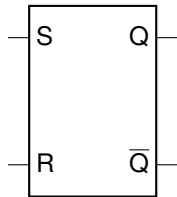
RS-Latch



R	S	$Q(t+1)$	$\neg Q(t+1)$	Verhalten
0	0	$Q(t)$	$\neg Q(t)$	Speichern
1	0	0	1	Rücksetzen
0	1	1	0	Setzen
1	1	?	?	?

NOR-SR-Latch

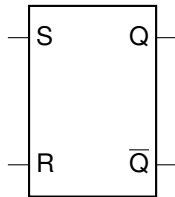
- pegelgesteuert
- **Set, Reset**
- „verbotener“ Zustand
 $(1, 1) \implies Q = \neg Q = 0$



S	R	Q
0	0	Q_{prev}
0	1	0
1	0	1
1	1	0

NAND-SR-Latch

- pegelgesteuert
- $\neg S, \neg R$
- „verbotener“ Zustand
 $(0, 0) \implies Q = \neg Q = 1$

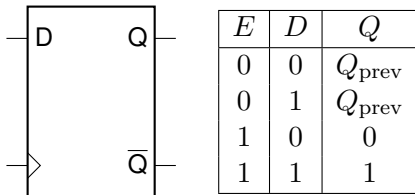


S	R	Q
0	0	1
0	1	1
1	0	0
1	1	Q_{prev}

Latches und Flipflops

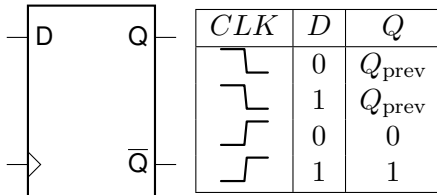
D-Latch

- pulsgesteuert
- transparent, wenn Schreibsignal aktiv ist
- Bei $E = 0$ bleibt Zustand gespeichert, sonst wird D übernommen.



D-Flipflop

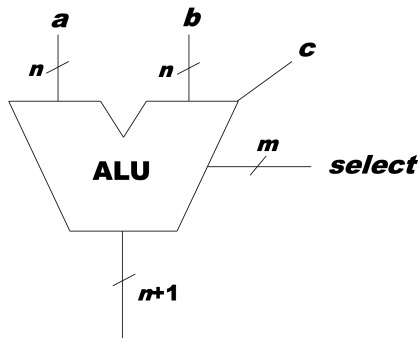
- taktf flankengesteuert
- Bei fallender Flanke bleibt Zustand gespeichert, bei steigender Flanke wird D übernommen.



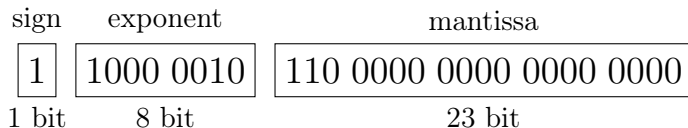
- **A**rithmetic **L**ogic **U**nit zur Berechnung von arithmetischen und logischen Basisoperationen

- n-Bit-ALU mit:

- ☐ zwei n-Bit-Operanden a, b, Eingangscarry c
- ☐ m-Bit select-Eingang, der auswählt, welche Funktion ausgeführt wird
- ☐ (n+1)-Bit-Ausgang

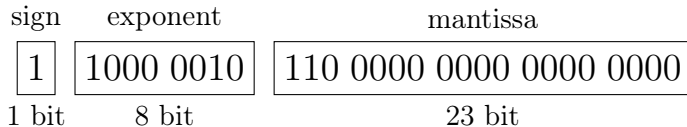


- Fließkommazahlen der Form $(-1)^{\text{sign}} \cdot 1.\text{mantissa} \cdot 2^{\text{exp}-\text{bias}}$



- Bei 32-Bit-Floats: 1 Bit Vorzeichen, 8 Bit Exponent, 23 Bit Mantisse, Bias 127
- implizite 1 vor der Mantisse wird nicht mitgespeichert
- Sonderfälle ± 0 , $\pm \infty$, NaN: nicht relevant für HA
- Visualisierung: [Float Toy](#)

Floating-Point-Zahlen: Beispiel



1. Vorzeichen: $(1)_2 \rightarrow (-1)$
2. Exponent: $(1000\ 0010)_2 = 130$, $130 - \text{bias} = 130 - 127 = 3$
3. Mantisse: $(1.110\ 0000\ 0000\ 0000\ 0000)_2 = 1.75$

$$n = (-1) \cdot 1.75 \cdot 2^3 = -14$$

Fragen?

Bis zum nächsten Mal ;)

Folien inspiriert von Niklas Ladurner und Prof. Dr. Robert Wille